

# **Everyday Practices of Agile Software Developers**

**Julia Prior**

**A Thesis submitted for the Degree of Doctor of Philosophy  
in Computing Sciences**

**Faculty of Engineering and Information Technology  
University of Technology, Sydney**

**July 2011**

# **Certificate of Authorship/Originality**

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student

---

# Acknowledgments

Producing a doctoral thesis is not an endeavour that I could have accomplished alone. Many people contributed to the effort in myriad ways. To the numerous friends, colleagues and family members who cheered me on from the sidelines and provided other valuable support, I thank you. While it is not feasible to acknowledge everyone, there are some contributions that must be specifically mentioned.

Firstly, an enormous thank you to the participant developers at Raptor who so generously allowed me into their working lives, as a stranger and novice, enthusiastically and patiently sharing their daily experiences with me. The CEO was exceptional in his enthusiasm, support and generosity for this research, giving me access to whomever and whatever I deemed necessary to carry out my fieldwork. I am extremely grateful to them all. I trust that the stories of their work in this ethnography and the findings of the research will make them proud to have played such a crucial part.

I was privileged to share the joys and despairs of the PhD journey with several exceptional students: Thomas Given-Wilson offered lively intellectual debates and challenging critiques of several ideas and chapter drafts; fellow doctoral candidates in the IDHuP research laboratory, Lian Loke, Penny Hagen, Kirsten Sadler, Astrid Larssen and Dean Hargreaves, in particular, helped me work through new ways of thinking about knowledge and our understanding of the world, as well as giving pragmatic moral support at pivotal moments; and my dear friend, Gordana Culjak — as we travelled the part-time PhD journey together over several years, albeit in very different types of research, her understanding, encouragement and advice helped me over many a hump in the road. I thank you all.

I also thank the other members and associates of the IDHuP lab for contributing to and being part of an holistic research space. I thank Tim Mansfield for his role as my acting supervisor in the middle of my candi-

---

dature particularly, but also for his advice, inspiration and example over many years. Thank you, too, to Ken Dovey, for his valued advice and encouragement throughout this project.

Richard Raban, former Head of the School of Software, viewed the completion of my thesis as vital for my own professional and personal development, as well as to the school and the faculty. My appreciation to him for continually insisting that it was a priority for me and providing managerial support and space to facilitate this project is immeasurable. Thank you.

I regard it as a tremendous honour to have been supervised in this research by two remarkable people. Toni Robertson, my principal supervisor, turned my view of the world upside down and inside out. She inspired and challenged me with interesting, but often daunting, suggestions and ideas to grapple with and explore, in order to engage more profoundly with the world's elusive, dynamic and intriguing complexities. And she never let me forget the fundamental importance to research rigour and integrity of the researcher's own critical, explicit positioning and accountability.

John Leaney, my mentor and co-supervisor, enthusiastically and generously embraced my proposed research direction from the first. His keen interest and belief in the value of the research never waned and set me back on track with renewed purpose, time and again.

Toni and John opened up alternative approaches to research in general, and software engineering practice in particular, that I had either not previously been exposed to or had not realised were scholarly options. They were dedicated to and engaged with me in this research process at a level that seems to be only dreamed of by most PhD students for their supervision. Thank you both for your commitment, your example and your unflagging belief in me.

My beloved parents, Mike Coleman and Niki Sampson, instilled in me from an early age an insatiable curiosity about the world, a love of study and a passion for the written word, thus preparing me for this particular learning journey. I am very grateful to my mother for casting her careful and professional eye over two drafts of my thesis, and for all her helpful suggestions.

---

My precious sons, Matthew and Luke, despite waiting patiently for years of their childhoods for bigger slices of my time and attention, showed great pride in my efforts throughout. Thank you both for providing perspective and laughter at all the right moments.

My husband and hero, Andrew, supported me through this endeavour for much longer than anyone should be expected to do. Throughout, he believed that it was only a matter of time before I was done, and he cheerfully shouldered the bulk of the domestic load to enable me to study. Whatever I say here to express my appreciation to him will be completely inadequate, so I will simply say thank you, from the bottom of my heart.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Research Interest and Questions . . . . .	4
1.2	Field Site for the Research . . . . .	6
1.3	Research Approach . . . . .	7
1.4	Research Findings . . . . .	9
1.5	Contributions of the Thesis . . . . .	11
1.6	Reading Matters . . . . .	12
1.7	A Guide to the Thesis . . . . .	12
1.8	Publications . . . . .	15
<b>2</b>	<b>Literature Review</b>	<b>16</b>
2.1	Prevalent SE Research . . . . .	17
2.1.1	Software Process Research . . . . .	17
2.1.2	Cognitive Research on Software Developer Behaviour	20
2.1.3	Empirical Software Engineering Research . . . . .	21
2.1.4	Information Systems Research into Software Engineering . . . . .	25
2.2	What's Missing in Prevalent SE Research . . . . .	27
2.2.1	Software Design as a Rational Process . . . . .	29
2.2.2	Software Development as a Cognitive Process . . . . .	30
2.2.3	Software Process in Retrospect . . . . .	32
2.2.4	The Grand Theory of Software Engineering . . . . .	33
2.2.5	So, What Needs to be Done? . . . . .	36
2.3	Work Practice Research . . . . .	40
2.3.1	HCI and CSCW Work Practice Research . . . . .	40
	Representations of Work Practice . . . . .	41
2.3.2	Ethnographic Studies of Professional Practice . . . . .	44
	Exemplar Ethnographic Studies . . . . .	45
	Ethnographic Studies and Software Engineering . . . . .	47
2.4	Agile Software Development . . . . .	50

---

2.4.1	What is Agile Software Development? . . . . .	50
2.4.2	Empirical Research on the Agile Approach . . . . .	51
2.4.3	Agile and Interpretivist Research . . . . .	53
2.5	Summary . . . . .	54
<b>3</b>	<b>Theoretical Commitments</b>	<b>55</b>
3.1	Theoretical Underpinnings . . . . .	57
3.1.1	Situated Action . . . . .	59
3.1.2	Situated Knowledges . . . . .	61
3.1.3	Boundaries . . . . .	64
3.1.4	Located Accountabilities . . . . .	66
3.1.5	Agency . . . . .	68
3.2	Conceptual Tools . . . . .	70
3.2.1	Infrastructure . . . . .	70
	Characteristics of Infrastructure . . . . .	71
3.2.2	Invisible Work . . . . .	74
	Abstractions of Work . . . . .	75
	What is Missing . . . . .	77
	Articulation Work . . . . .	79
	Articulation Work and Software Development . . . .	82
3.3	Summary . . . . .	86
<b>4</b>	<b>Ethnography for Software Development Research</b>	<b>88</b>
4.1	Research Using Ethnography . . . . .	89
4.1.1	What is Ethnography? . . . . .	89
4.1.2	Why Use Ethnography? . . . . .	90
4.1.3	Theoretical Perspective and Philosophy for Ethnog- raphy . . . . .	91
4.1.4	Characteristics of Ethnography . . . . .	92
	In-Depth Sense-Making . . . . .	93
	Triangulated Data, Logic-of-justification . . . . .	94
	Reflexivity . . . . .	94
4.2	Ethnography and Culture . . . . .	95
4.3	What People Say They Do, What People Do . . . . .	96
4.3.1	Analysis of Cultural Behaviour . . . . .	97
	Patterns <b>For</b> Behaviour . . . . .	98
	Patterns <b>Of</b> Behaviour . . . . .	99
4.4	My Experience of Using Ethnography . . . . .	100
4.4.1	Thinking About Knowledge . . . . .	101
	Transformative Texts . . . . .	102
4.4.2	As Stranger and Novice . . . . .	104

---

4.4.3	Working in the Field . . . . .	106
	Tools of my Trade . . . . .	108
4.4.4	Data Management . . . . .	109
4.4.5	Reflection, Analysis and Writing . . . . .	112
	Ethnographic Data Analysis . . . . .	112
	Using Existing Theory . . . . .	113
4.5	Ethical Issues . . . . .	114
4.6	Summary . . . . .	116
<b>5</b>	<b>On Being a Raptor Software Developer</b>	<b>118</b>
5.1	The Fieldwork Setting . . . . .	119
5.1.1	The Physical Environment . . . . .	121
5.1.2	Organisational Culture . . . . .	122
5.1.3	Team Structures . . . . .	124
5.1.4	Raptor's Development Methodology . . . . .	125
5.2	A Raptor Developer's Working Life . . . . .	127
5.2.1	Episode One: Autumn . . . . .	129
5.2.2	Episode Two: Spring . . . . .	132
5.2.3	Episode Three: The Following Autumn . . . . .	135
5.3	Summary . . . . .	138
<b>6</b>	<b>Infrastructure for Raptor Software Developers</b>	<b>139</b>
6.1	Developers' Local Infrastructure . . . . .	141
6.2	Primary Tools . . . . .	143
6.2.1	Development Environment . . . . .	144
6.2.2	Version Control . . . . .	145
6.2.3	Test Harness . . . . .	147
	The CrikeyMonitor . . . . .	148
	Unit Tests . . . . .	149
6.3	Secondary Tools . . . . .	150
6.3.1	Software Architecture and Database . . . . .	151
	C-architecture . . . . .	151
	Database Software . . . . .	152
6.3.2	Workload and Task Management Tools . . . . .	153
	Connexion . . . . .	153
	Tracker . . . . .	156
6.3.3	Communication Tools . . . . .	157
	Email . . . . .	157
	Wiki . . . . .	158
6.4	Policies and Procedures . . . . .	158
6.4.1	Software Testing . . . . .	158



---

	Test-Driven Development and Test First . . . . .	159
6.4.2	Quality Assurance . . . . .	161
	Code Reviews . . . . .	161
	Check-in Scheduling . . . . .	162
	Coding Standards . . . . .	162
6.4.3	Workload Management . . . . .	163
	Work Items and Task Assignment . . . . .	163
	Team Stand-ups . . . . .	163
6.5	Summary . . . . .	164
<b>7</b>	<b>Infrastructuring the Code, Coding the Infrastructure</b>	<b>165</b>
7.1	Infrastructuring the Code . . . . .	167
	7.1.1 Code to Produce More Code . . . . .	167
	7.1.2 Infrastructure as Part of the Product Connect . . . . .	169
7.2	Coding the Infrastructure . . . . .	171
	7.2.1 Product Code as Infrastructure Code . . . . .	171
	7.2.2 Coding the Test Infrastructure . . . . .	174
	Changes to DAT and the CrikeyMonitor . . . . .	174
	7.2.3 Using Infrastructure to Develop More Infrastructure . . . . .	176
7.3	Infrastructure Work, Primary Work . . . . .	177
	7.3.1 Infrastructure as Primary Work . . . . .	178
	7.3.2 Work Required When Infrastructure Becomes Opaque . . . . .	179
	“Pretend this is true” . . . . .	179
	“You broke BuildIt!” . . . . .	180
7.4	Intertwining Infrastructure and Product . . . . .	184
	7.4.1 Infrastructure for Code . . . . .	184
	7.4.2 Code as Infrastructure . . . . .	185
	7.4.3 Infrastructure Work and Primary Work . . . . .	186
	7.4.4 Relations Between Infrastructure and Product . . . . .	186
	Technical Developers as Technical Users . . . . .	188
7.5	Summary . . . . .	190
<b>8</b>	<b>Mutually Constitutive Relations</b>	<b>192</b>
8.1	Test First at Raptor . . . . .	193
	8.1.1 Test First as a Pattern for Behaviour . . . . .	194
	8.1.2 Attitudes towards the Test First PfB . . . . .	196
	8.1.3 Test First in Practice . . . . .	198
8.2	New PfBs are Introduced . . . . .	205
	8.2.1 New PfBs in Practice . . . . .	208
	Code Reviews in Practice . . . . .	209
	Check-in Scheduling in Practice . . . . .	215

---

8.3	Relations of Infrastructure with Work Practice . . . . .	219
8.3.1	Practice Shaping Infrastructure . . . . .	220
8.3.2	Infrastructure Moulding Practice . . . . .	221
8.3.3	Shifting Boundaries and Agencies . . . . .	224
<b>9</b>	<b>The Importance of Studying Software Development as Situated Action</b>	<b>225</b>
9.1	Research Issues and Approach . . . . .	226
9.1.1	Infrastructure as a Central Issue . . . . .	228
9.2	Thesis Findings . . . . .	229
9.2.1	Characteristics of Infrastructure . . . . .	230
9.2.2	Infrastructure and Product Code . . . . .	232
	Infrastructure Made of Program Code . . . . .	232
	Coding the Infrastructure . . . . .	233
9.2.3	Infrastructure and Work Practice . . . . .	235
9.3	Thesis Boundaries . . . . .	238
9.4	Some Implications of Studying SWD Practice . . . . .	241
9.5	Future Work . . . . .	243
9.6	Thesis Contributions . . . . .	244
	<b>References</b>	<b>246</b>

# List of Figures

4.1	Interaction Map used to Record Developer Interactions . . .	109
6.1	Raptor Infrastructure . . . . .	143
6.2	Primary Software Development Tools . . . . .	144
6.3	The CrikeyMonitor Webpage . . . . .	148
6.4	Image Shown When a Crikey! Occurs . . . . .	149
6.5	Secondary Software Development Tools . . . . .	150
6.6	Policies and Procedures . . . . .	159
7.1	Changes to Incident Manager . . . . .	172

## Abstract

The theory and practice of software engineering is largely represented in the literature by discrete rational models, methodologies, tools and techniques for producing software and for describing work processes as they ought to be done. There has been little research that characterises software development as it is actually practised every day by professional software developers. This thesis addresses that gap.

As the recognition of software development as embodied work is fundamental to this research, a situated approach underpins its exploration of software development as it is performed by real developers in their workplace at the time that it occurs. The initial, guiding research question, *“So what **do** software developers do every day?”* was addressed by undertaking a longitudinal ethnographic study, involving two years of fieldwork, of an Australian group of software developers who use an Agile development methodology.

A local software development infrastructure, comprising software tools and formal software development policies and procedures, is available to the developers. As the research progressed, it became obvious that they spent an enormous amount of their time and effort dealing with this infrastructure in order to accomplish their daily work. Further, more directed questions about the developers’ daily practice thus became distilled into this one:

*What significance does their infrastructure have for the developers’ everyday work?*

The findings challenge two assumptions often implicit in software engineering research interests: firstly, that infrastructure, along with its constituent elements, is a given — a discrete, independent and stable entity — and, secondly, that software development practice can be authentically represented by, and understood in, formal, rational models of software process. The foregrounding of relations between the developers, their infrastructure and their work product forms the crux of this thesis. Its primary contributions are:

- It presents an ethnography of a little-studied group: professional software developers;

- It provides a detailed account of how the developers' infrastructure and their work practice are intimately intertwined and mutually constitutive, continuously generating the context in which the design and development of complex software can be accomplished;
- It articulates the processes and practices that effect progressive blurring and co-dependency between the software product and the infrastructure;
- It identifies and describes essential dimensions of developers' practice that are missing from accounts and representations such as formal models;
- It demonstrates how an explication of local infrastructure is more beneficial to understanding specific software development work practice than abstract definitions of infrastructure which are not constrained by the practicalities of everyday use.